ActiveState®

How to Manage Risk of Your Polyglot Environments

Presenters

- **Jeff Rouse**, VP Product, ActiveState
- **Pete Garcin**, Senior Product Manager, ActiveState
- **Larry Maccherone,** Head of DevSecOps Transformation, Comcast

ActiveState®

VP Product

Jeff Rouse, ActiveState

Manage Risk:
Polyglot Environments

Jeff Rouse
VP Product
ActiveState

ActiveState

Manage Risk: Polyglot Environments

Track-record: 97% of Fortune 1000, 20+ years open source

Polyglot: 5 languages - Python, Perl, Tcl, Go, Ruby

Runtime Focus: concept to development to production

ActiveState

Manage Risk: Polyglot Environments

What is Polyglot?

SQL

ActiveState

How Do Polyglot Environments Evolve?

- **Technology.** Best tool for the job, modern software projects.
- **People.** technology stacks added through acquisition, changes in tech leadership
- **Time.** technologies come in & out of favour; old languages never die.

Manage Risk: Polyglot Environments

Every Organization is Polyglot

- Any desktop application with an online component.
- YAML configuration used with any project.
- An application with embedding scripting.

ActiveState

Manage Risk:
Polyglot Environments

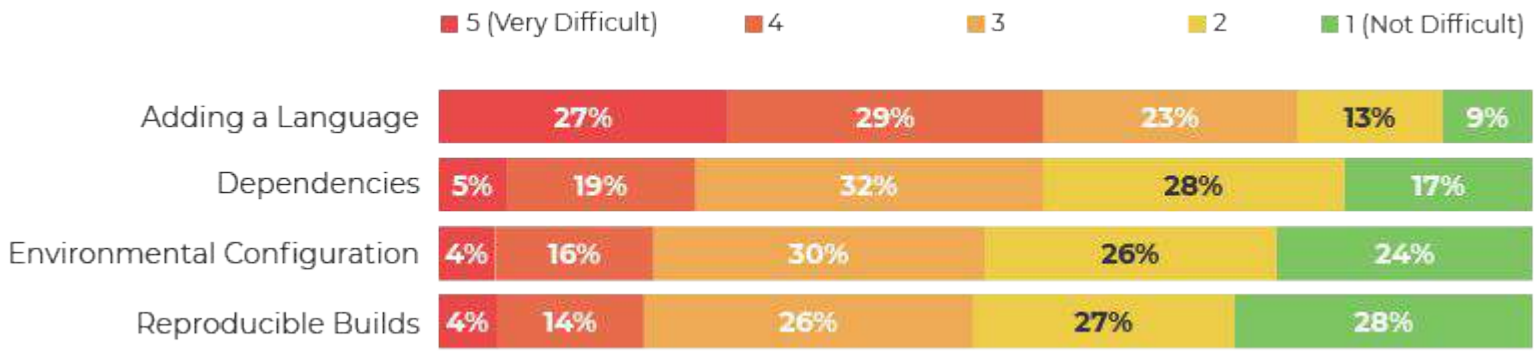
Adding a Language

Source: ActiveState Developer Survey 2018, Open Source Runtime Pains

ActiveState®

Manage Risk: Polyglot Environments

Rank the Challenges

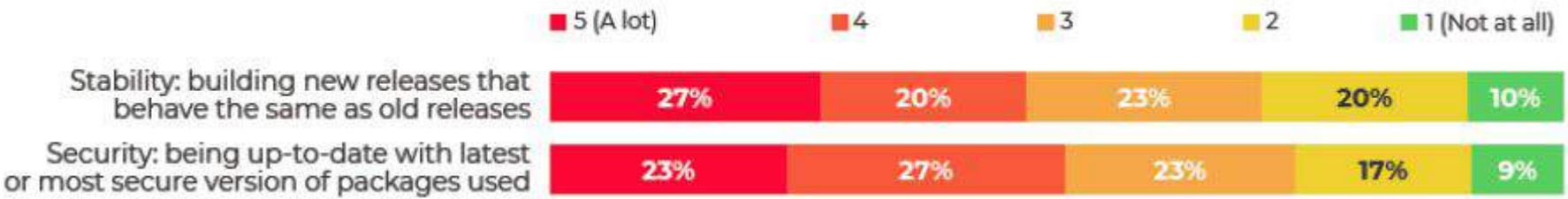


Source: ActiveState Developer Survey 2018, Open Source Runtime Pains

ActiveState

Manage Risk: Polyglot Environments

Stability & Security → Painful

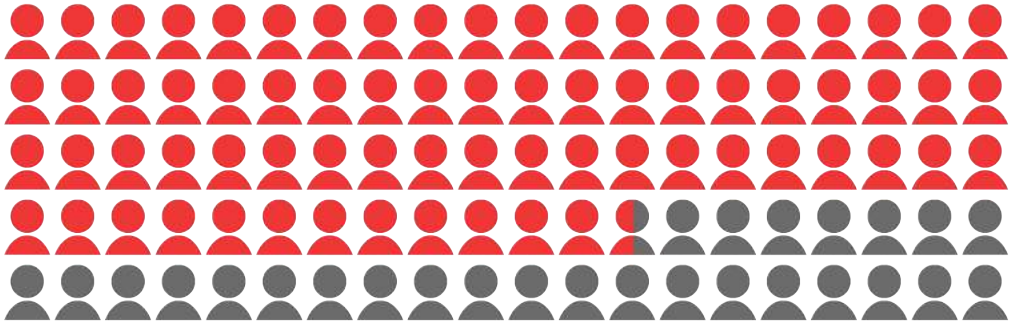


Source: ActiveState Developer Survey 2018, Open Source Runtime Pains

ActiveState

Manage Risk: Polyglot Environments

Hidden Costs



75%

Managing dependencies

Source: ActiveState Developer Survey 2018, Open Source Runtime Pains

ActiveState®

Benefits

- **Speed.** Ship faster: better products, better innovation.
- **Recruitment.** Be attractive workplace: enable coders to choose the tools they need.

Drawbacks

- **Variability.** Tooling support & programming language quality.
- **Expertise Gap.** Deep core competency at odds with breadth of programming languages.
- **Dependencies.** Larger pool of dependencies.
- **Support Costs.** Unable to centralize, maintenance.

Magnified Issues

How will you monitor, identify and resolve?

Production bugs, Common Vulnerabilities & Exposures (CVE), threats; additional risk exposure with 3rd party dependencies.

Equifax Breach: out of date 3rd party dependency

Presentation Title

Resolutions

Reduce Libs

Reduce Tools

Reduce Services

Reduce Attack Surface

Robust Processes, Automated and Centralized for Visibility

ActiveState

ActiveState®

Senior Product Manager

Pete Garcin, ActiveState

Manage Risk:
Polyglot Environments

Pete Garcin

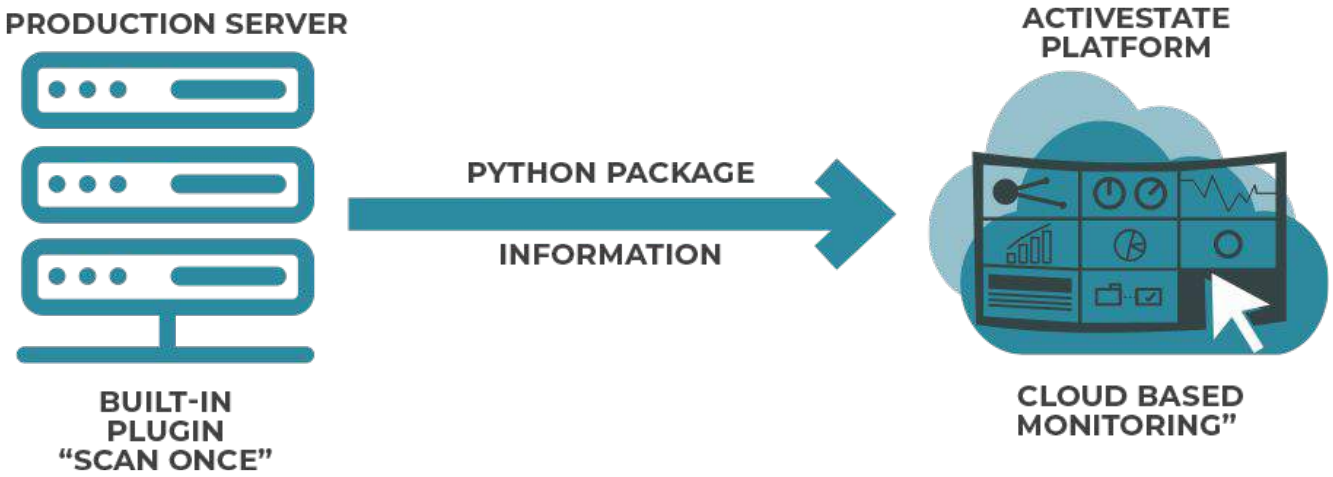
Senior Product Manager

ActiveState

ActiveState

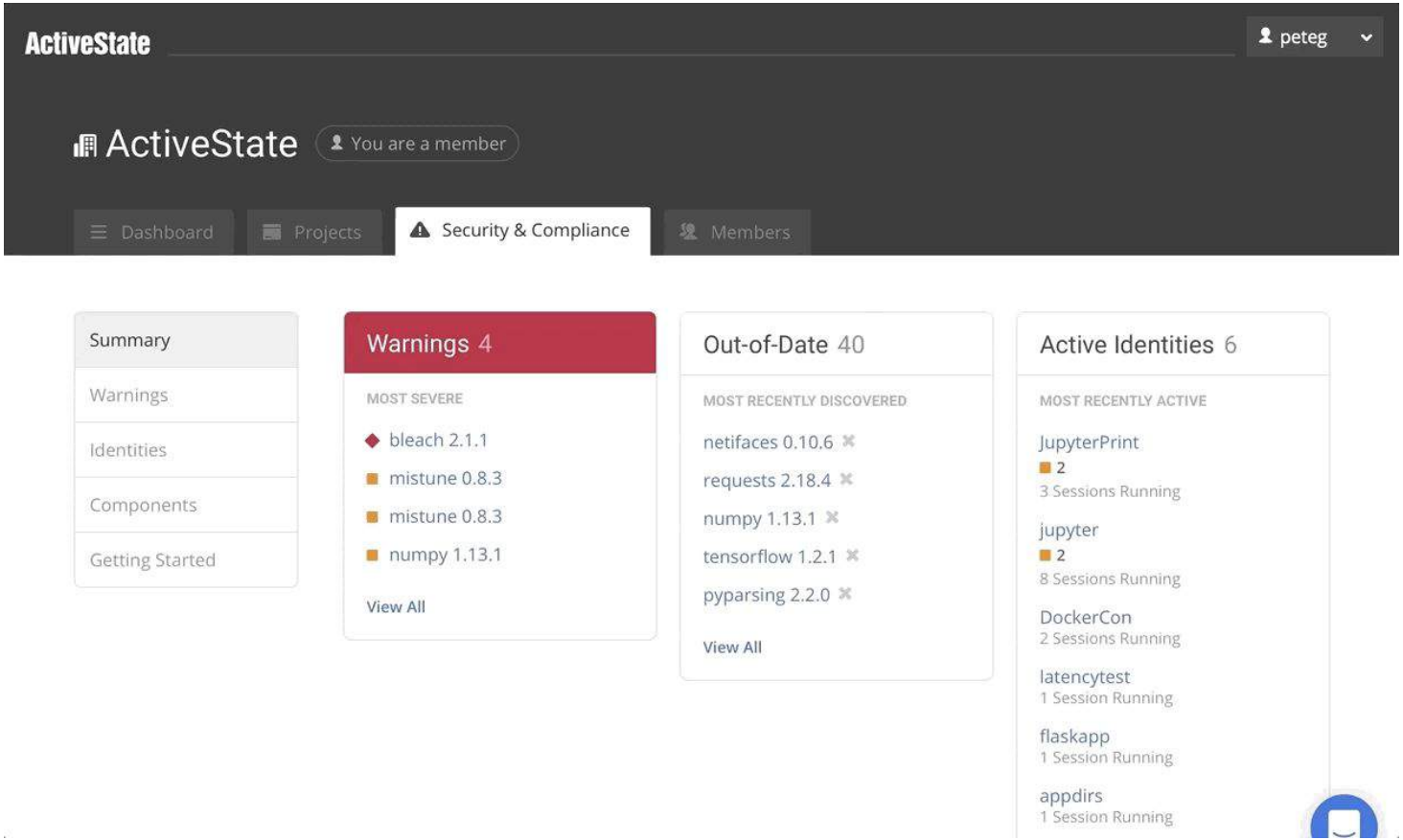
Manage Risk: Polyglot Environments

Automated Processes



ActiveState

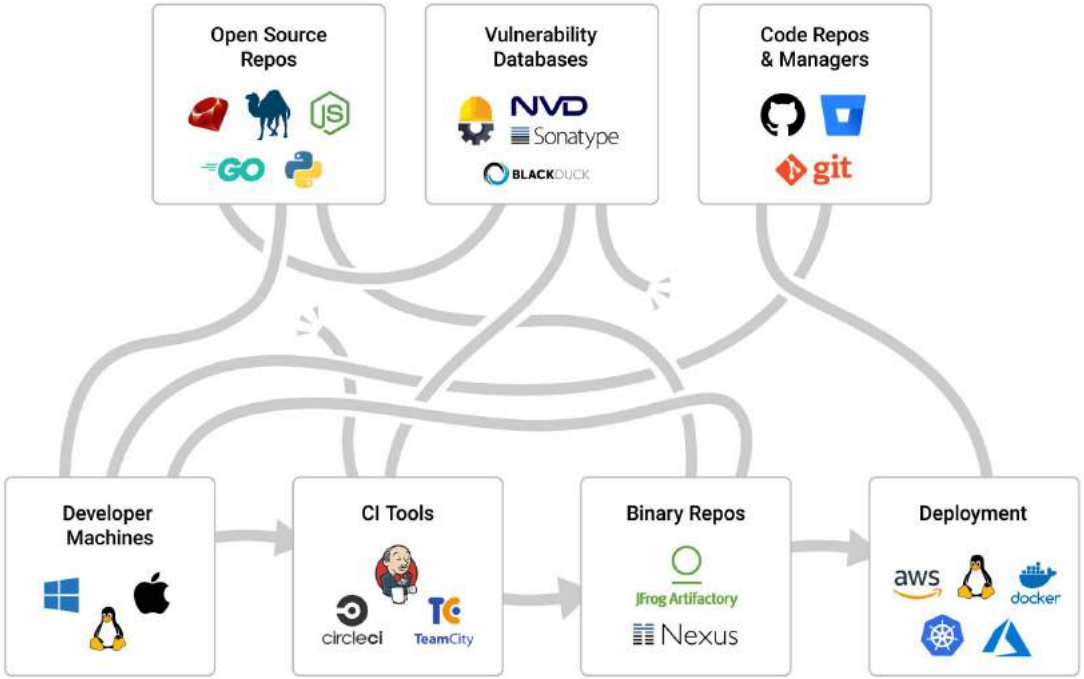
Manage Risk: Polyglot Environments



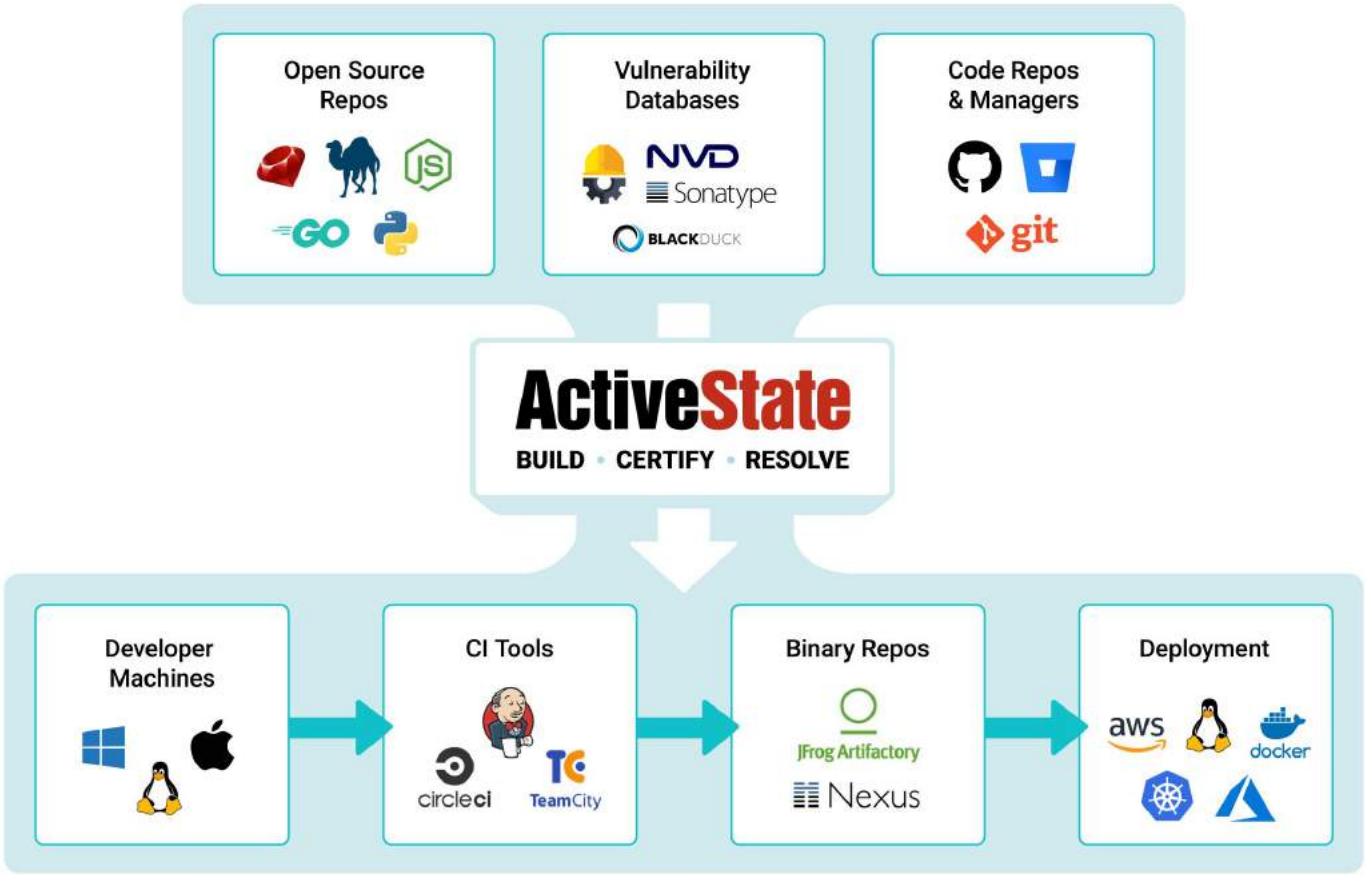
Automating Environments

- **Automate.**
- **Bundle.**
- **Simplify Shares.** Encourage adoption of common environments.

Manage Risk: Polyglot Environments

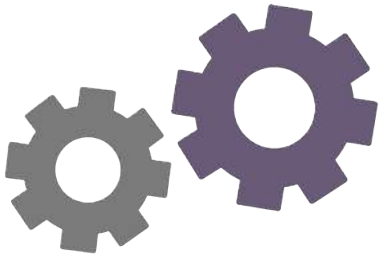


Manage Risk: Polyglot Environments



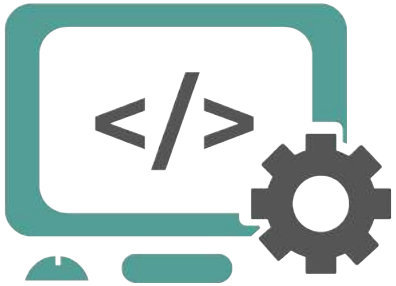
Manage Risk: Polyglot Environments

Solving Core Problems



Environment Configuration

Dependency Management



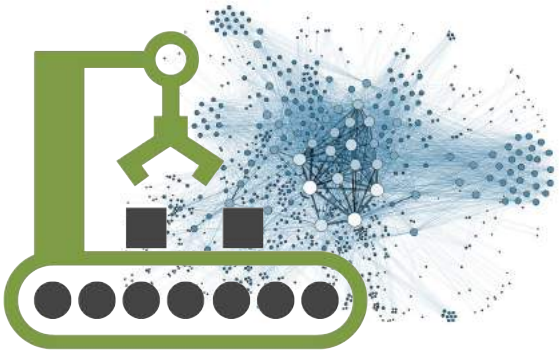
Workflow Configuration

Manage Risk: Polyglot Environments

Best Practices - Build Eng & Development

Shrink Build

Build Standard



Reproduce & Manage

ActiveState

Best Practices - Development to Production

Monitor Runtime

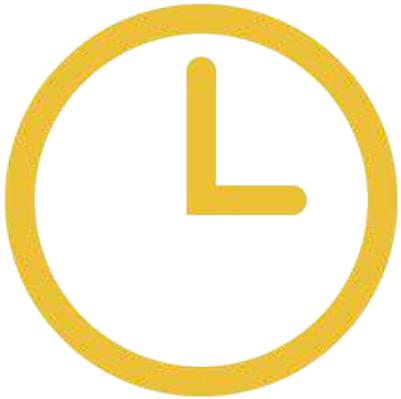
Get Updates

Manage Risk: Polyglot Environments

Benefits to You

Dev Zen

Same Same



Time

ActiveState

Security at the Speed of Software Development

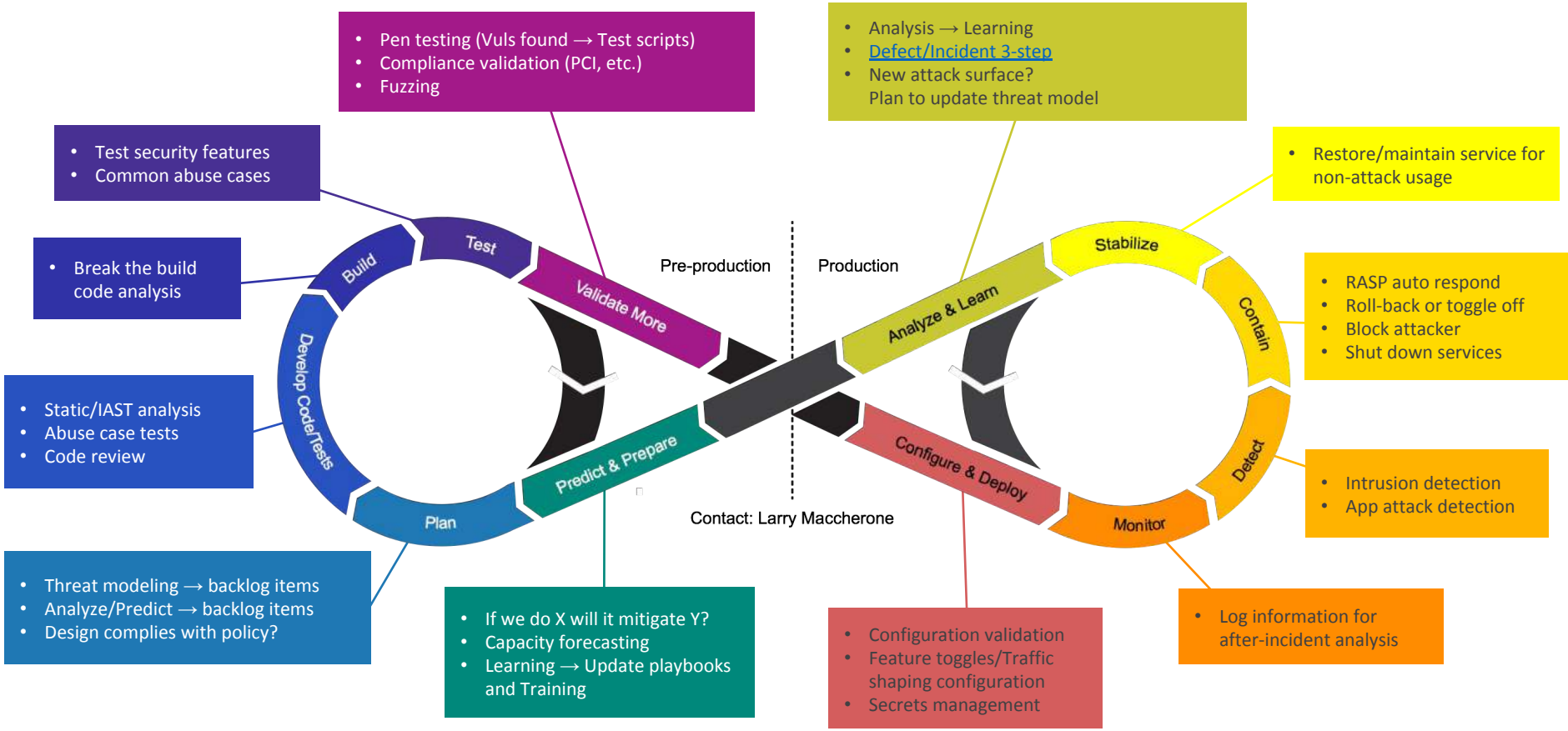
A lean/agile transformation approach to achieving a DevSecOps culture

Presented by:
Larry Maccherone

COMCAST

Larry Maccherone
LinkedIn.com/in/LarryMaccherone

Security practices on DevOps continuum ➔ DevSecOps



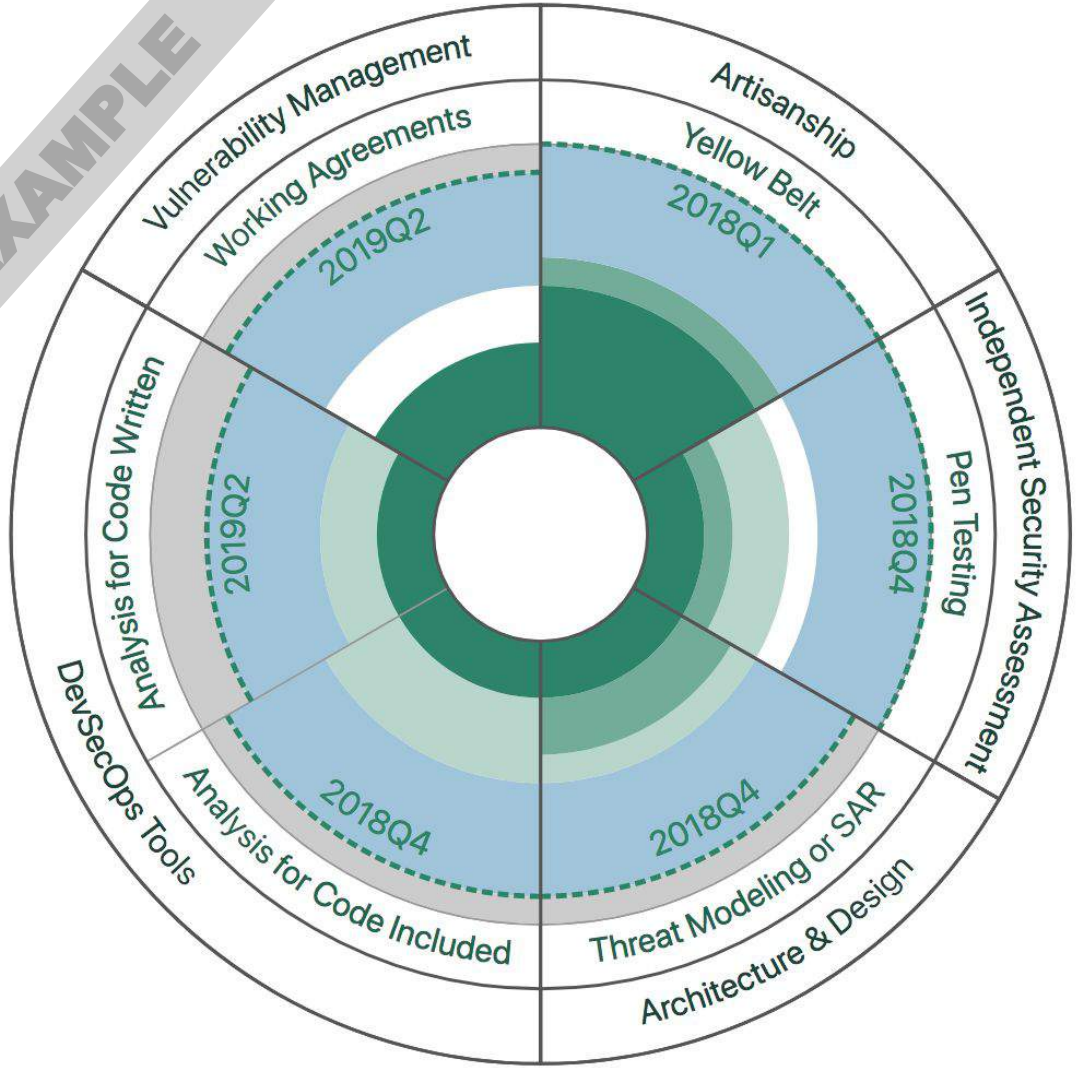
That’s a lot of stuff!

How do we get
development teams to adopt?

Disciplines	← Best bang-for-the-buck (effort)			More expensive/harder →
Artisanship	100% of group members who have been employed at least 90 days have been through **Yellow Belt** training	Group has access to at least one **Green Belt**	Group has: • Minimum 15% Green Belt • At least one **Brown Belt**	Group has: • Minimum 25% Green Belt • Minimum 5% Brown Belt • At least 1 **Black Belt**
Architecture and Design	As appropriate, a **threat model** or a security architecture review (**SAR**) has been done and kept up to date with every significant change to the attack surface and resolved all issues within SLA (typically 120 days for critical and high)			
DevSecOps Tools	Software Composition Analysis (SCA, **analysis for code included,** aka open source security) tool(s) is integrated into the pipeline and auto run on commit, merge, or cadence that is significantly shorter than the release cadence	Primary Code Analysis (PCA, **analysis for code written,** typically SAST or IAST) tool(s) is integrated into the pipeline and auto run on commit, merge, or cadence that is significantly shorter than the release cadence	**Fuzzing** (black-box dynamic) run in CI or on cadence and all scans clean	Clean white box dynamic (Web application dynamic scanning, App-server instrumented scanning, etc.) results
Vulnerability Management	Team has "**working agreements**" on how it becomes aware of, triages, and resolves findings in the current dev cycle to stay within current team policy (see "Your Team Vulnerability Policy" below). Ex: Notices in Slack/email; and/or findings put in Jira backlog via integration and considered at planning; and/or some highly visible console, such as the Findings Metrics Portal (or equivalent) is checked on a regular cadence; and/or pipeline is interrupted ("failed build" or disabled merge button), etc. Sources of vulnerabilities might include SCA/PCA tools, network-originated scans, independent security assessment, threat modeling, and fuzzing			For any non-false-positive finding that escapes the current development cycle and is being reported externally (e.g. ServiceNow), a formal system (default CVSS v2.0 base) has been used to "score" the vulnerability.
Your Team Vulnerability Policy	"**Stop the bleeding**" - For iss…nd by the team's own tools or practices the policy … interrupt the pipeline or stop progress u… assuring that no **new** (not … high critical vulnerabilities esc… rent dev cy…	All **critical and high** severity findings (if available, CVSS 2.0 >= 7.0) are resolved (fixed, marked false positive, or risk accepted) in the current dev cycle for issues found by the team's own tools and practices and within 120 days if …ed from outside the team	All critical, high, and **medium** severity findings are resolved (fixed, marked false positive, or risk accepted) in the current dev cycle for issues found by the team's own tools and practices and within 120 days if received from outside the team	All critical, high, medium, and **low** severity findings are resolved (fixed, marked false positive, or risk accepted) in the current dev cycle for issues found by the team's own tools and practices and within 120 days if received from outside the team
Network-originated Scans	Findings for **network-originated scans** (ex: …) against your components by the security team are re… the SLA (typically 120 days for critical and high)	…ux a… …s servers that our applications run …inc… …date at appropriate …ates can …	**Authenticated automated self-scans** (UScan, etc.) are run and findings are resolved in the current dev cycle	(PCI only) **Authenticated scans** are being run for your components and findings resolved within SLA
Independent Security Assessment	Independent Security Assessment (aka **Pen testing**) done kept up to date at appropriate cadence (e.g. m… and issues r… within SL…			**Red/Purple Team** exercises done on the system your solution is part of and issues resolved in the SLA
Secrets Management		A **secrets management** system or some other approach … used so that no credentials need to be put into source code repositories	… blocks the commit unless it is free of creden…	
Asset Management	All of your 'Active' applications and their components are represented in **ITRC** or ServiceNow Innovations Operations Platform (**IOP**) as appropriate	In ITRC all components of your applications have a **Security POC** listed that is best suited to respond properly to a security threat (To find: Application > Component > Responsibilities)		
Reference Components and Designs		Mostly built with **reference components and designs,** such as those in Secure Design Patterns	Non-reference components have had a **security code review**	Team creates new secure components and designs, ensures that they are properly vetted, and then **contributes** them to the greater Comcast community
Incident / Public Vulnerability Response Capability	Team **knows how and when to contact:** • The Security Fusion Center for incidents (The Security Response Center 1-877-249-7306 (SRC is the always manned portion of the SFC) • …for Public, External vulnerabilities	**Published,** detailed **Playbook,** consisting of: • Up-to-date roles with 24*7*365 contacts • Workflows • Prioritization / Scoring / Ranking • Tracking Repositories • Working Agreements	Playbooks tested and proven through detailed **tabletop exercises** or "hands-on" war-gaming.	Playbooks tested and proven through "chaos" (surprise) **war-gaming**, "simulated" attack (executed by existing security team members), or an actual external attack.
	Others TBD including 1) Network (zones replacement), 2) Cloud and containers, 3) Abuse cases, 4) Ops visibility – detect attacks in progress and push patches within 24 hours			

EXAMPLE

Visualizing an Org's practices



Culture	We have fully adopted this practice
Actions	We're in the process of adopting this practice
Words	We're making plans to adopt this practice
Thoughts	We do not have plans for this practice
Unknown	Unassessed or Needs follow up
Trade-off	This practice is not worth it in this context

Dev*[Sec]*Ops is…
empowered engineering teams
taking ownership
of how their product
performs in production
[including security]

DevSecOps Manifesto

Build security in
more than bolt it on

Rely on empowered engineering teams
more than security specialists

Implement features securely
more than security features

Rely on continuous learning
more than end-of-phase gates

Build on culture change
more than policy enforcement

We, the Security Team…

Recognize that Engineering Teams…

- Want to do the right thing
- Are closer to the business context and will make trade-off decisions between security and other risks
- Want information and advice so those trade-off decisions are more informed

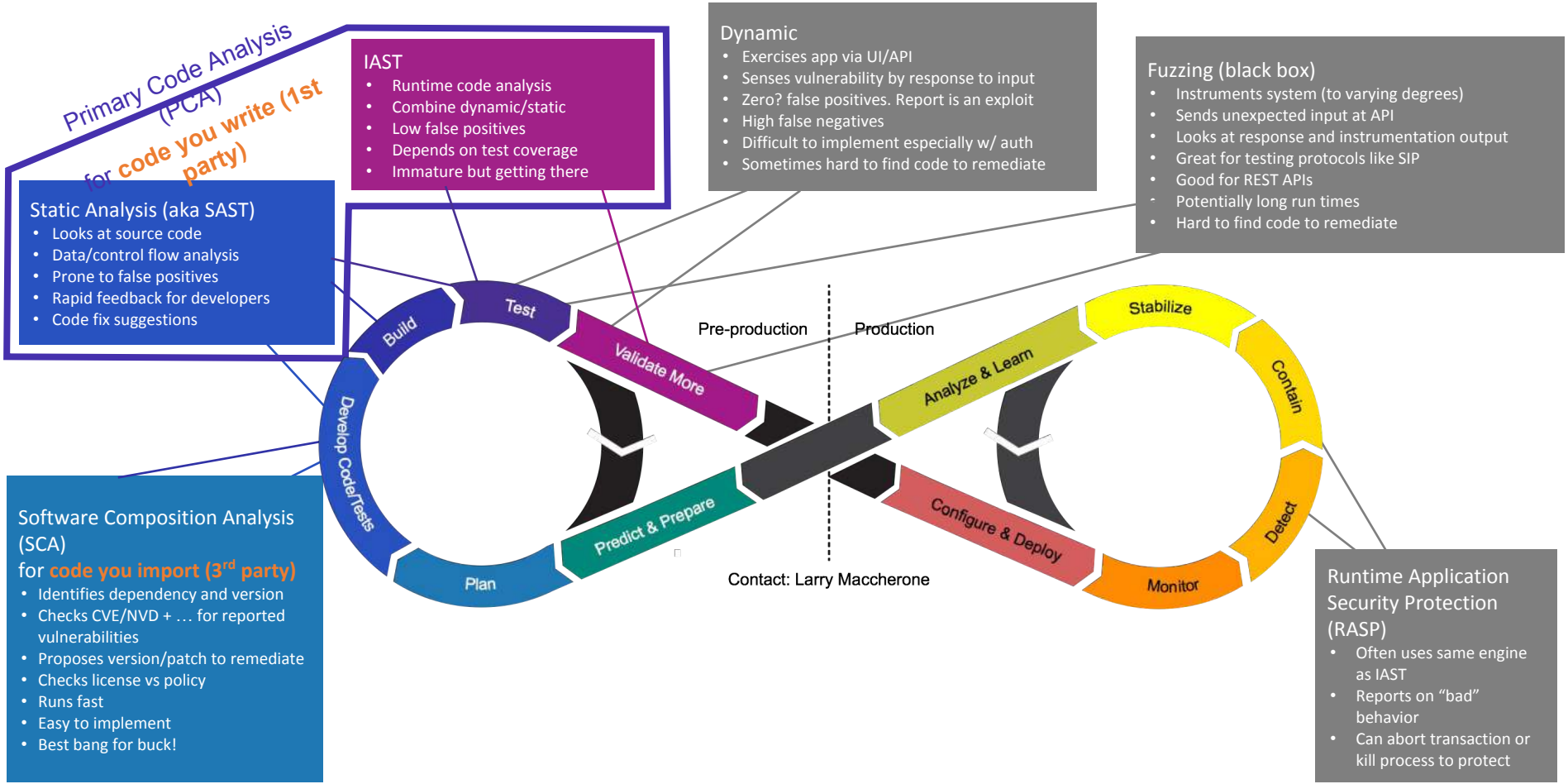
Pledge to…

- Lower the cost/effort side of any investment in developer security tools or practices
- Assist 2x as much with preventative initiatives as we beg for your assistance reacting to security incidents

Understand that…

- We are no longer gate keepers but rather tool-smiths and advisors

DevSecOps Tool Landscape



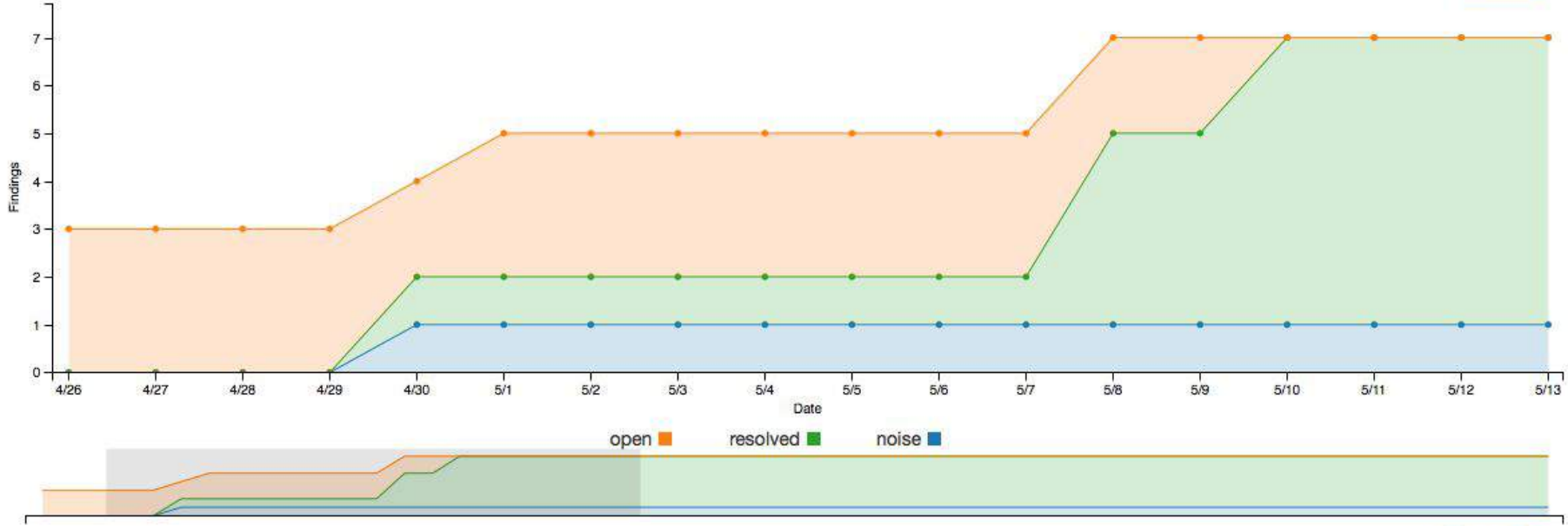
Metrics Portal

DevSecOps

FINDINGS ADOPTIONS ADMIN LOGOUT

Finding Metrics

devsecops | Sources | ALL | HIGH | ALL | Origin | Submit



v0.9.13

What's next?

- Questions?
- Pilot this DevSecOps transformation framework with a few of your teams
- Connect with me on: LinkedIn.com/in/LarryMaccherone

Q & A

ActiveState®

What's Next

- Watch a demo: https://www.youtube.com/watch?v=c5AIxN9ehrI
- Get a demo marketing@activestate.com
- Contact us for the language build you need: platform@activestate.com

ActiveState®

Where to find us

Tel: **1.866.631.4581**

Website: **www.activestate.com**

Twitter: **@activestate**

Facebook: **/activestatesoftware**

ActiveState®